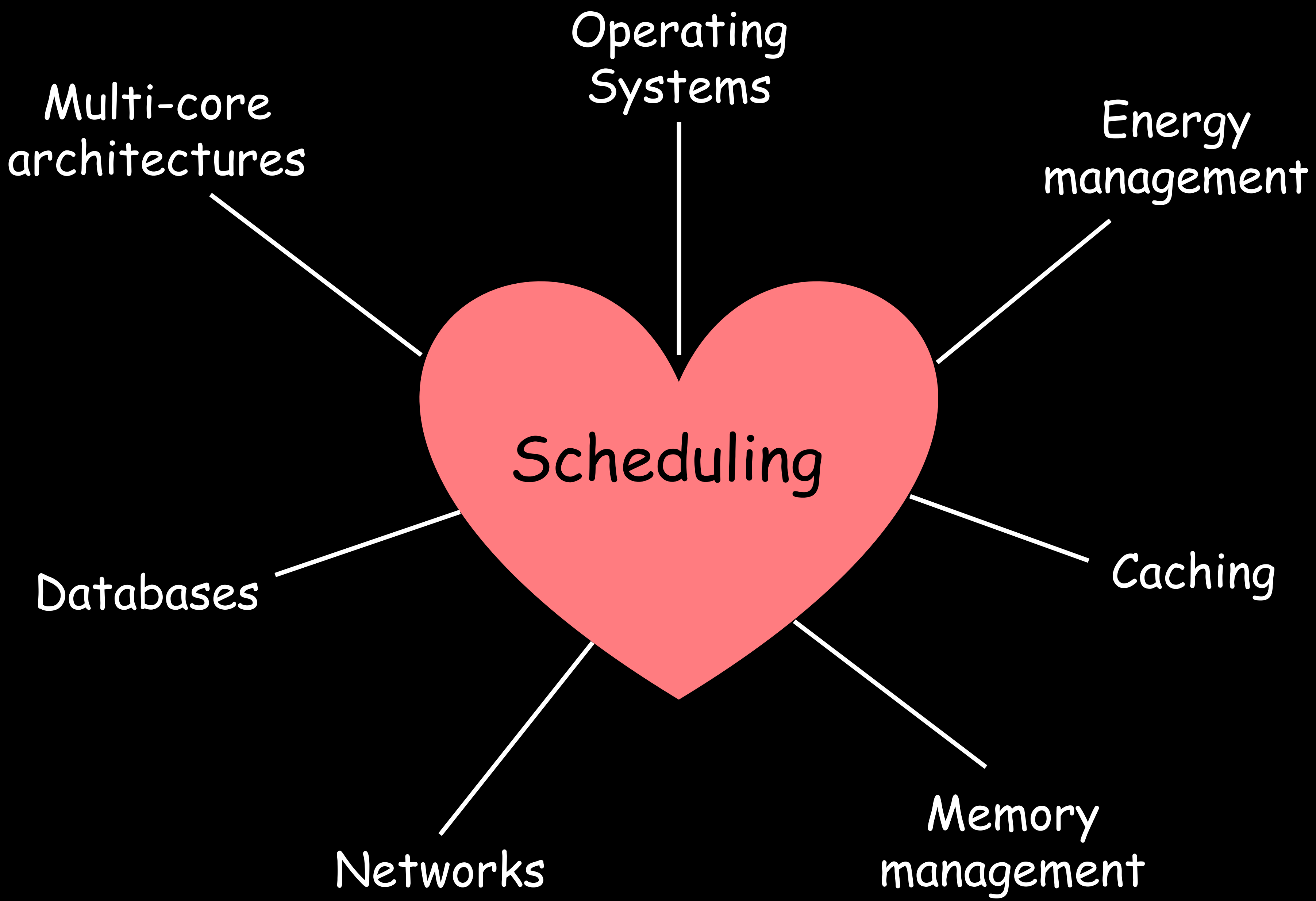
Recent Breakthroughs in Stochastic Scheduling Theory

Mor Harchol-Balter
Computer Science Dept.
Carnegie Mellon University

schedulingseminar.com


Operating Systems
Multi-core architectures
Energy management
Scheduling
Databases
Caching
Networks
Memory management

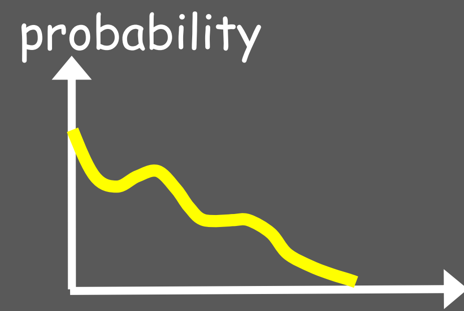
Scheduling

Worst-case

Adversary chooses job sizes

Adversary chooses arrival times

Stochastic

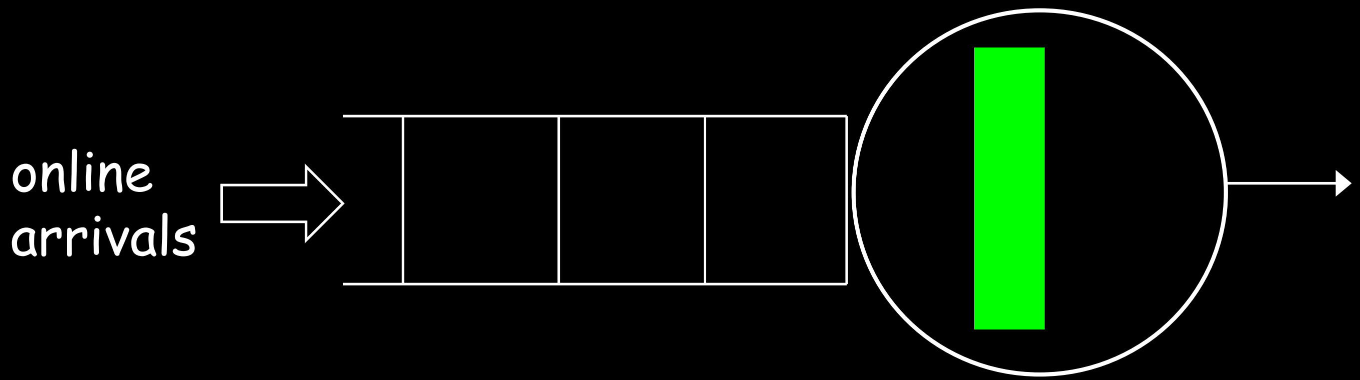


Job sizes drawn from distribution

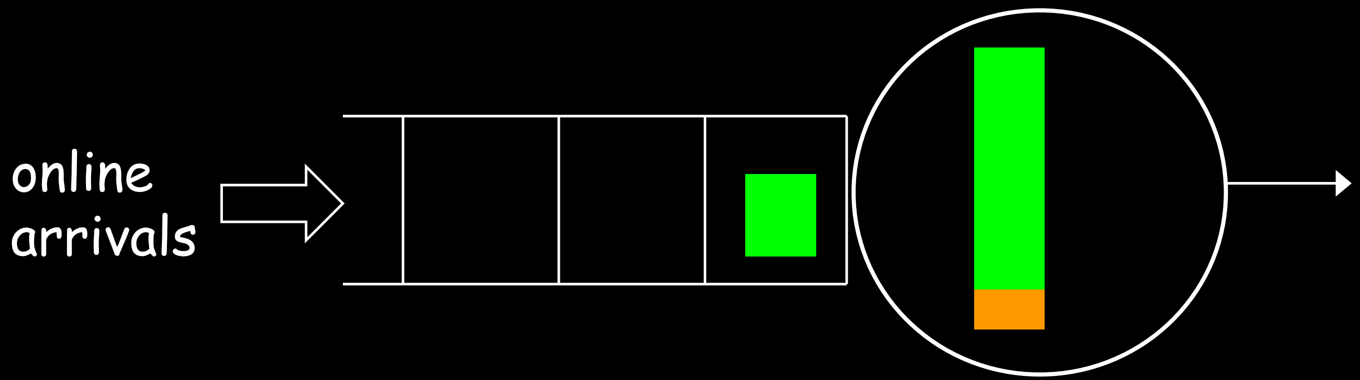
Arrival times drawn from distribution

Some folks in Stochastic Scheduling

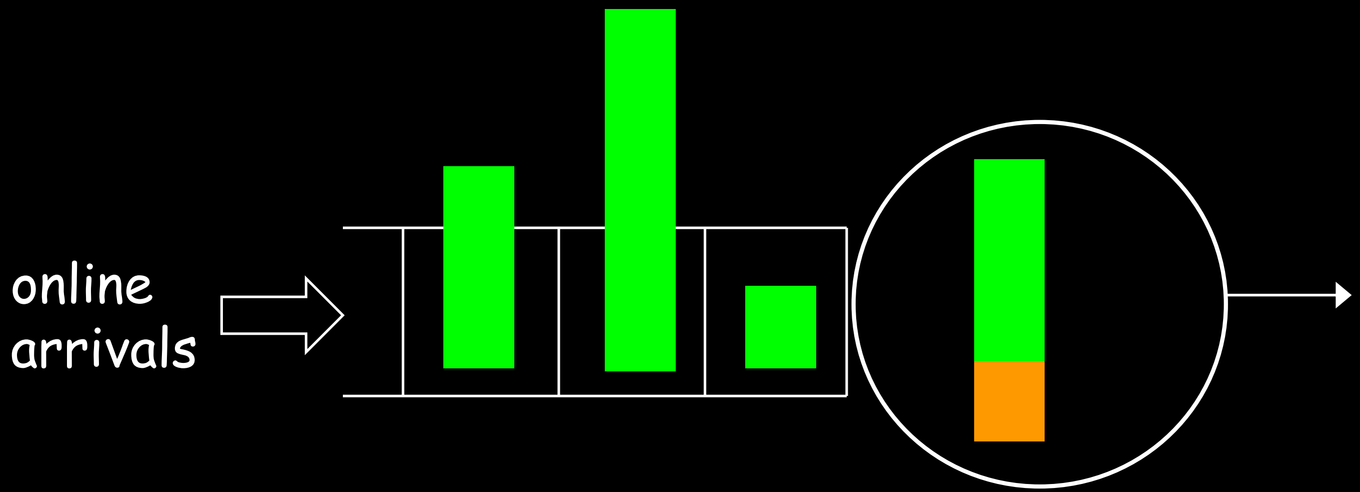
Basic Terminology



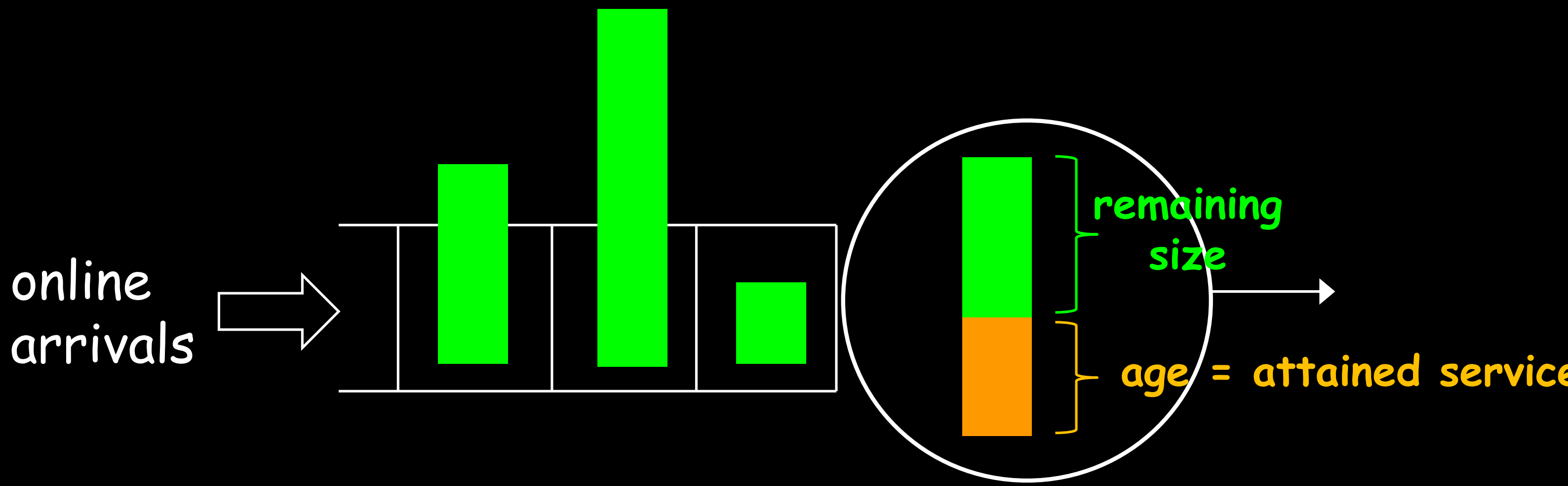
Basic Terminology



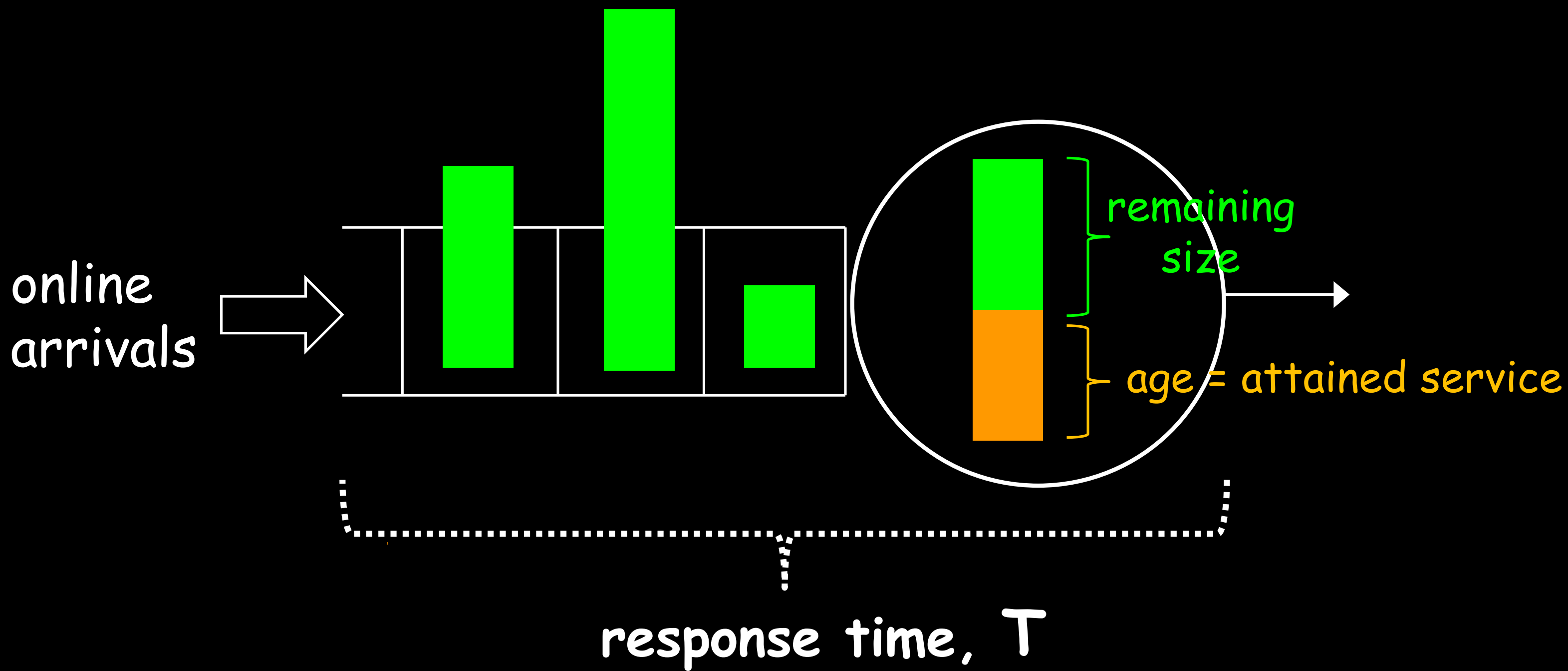
Basic Terminology



Basic Terminology

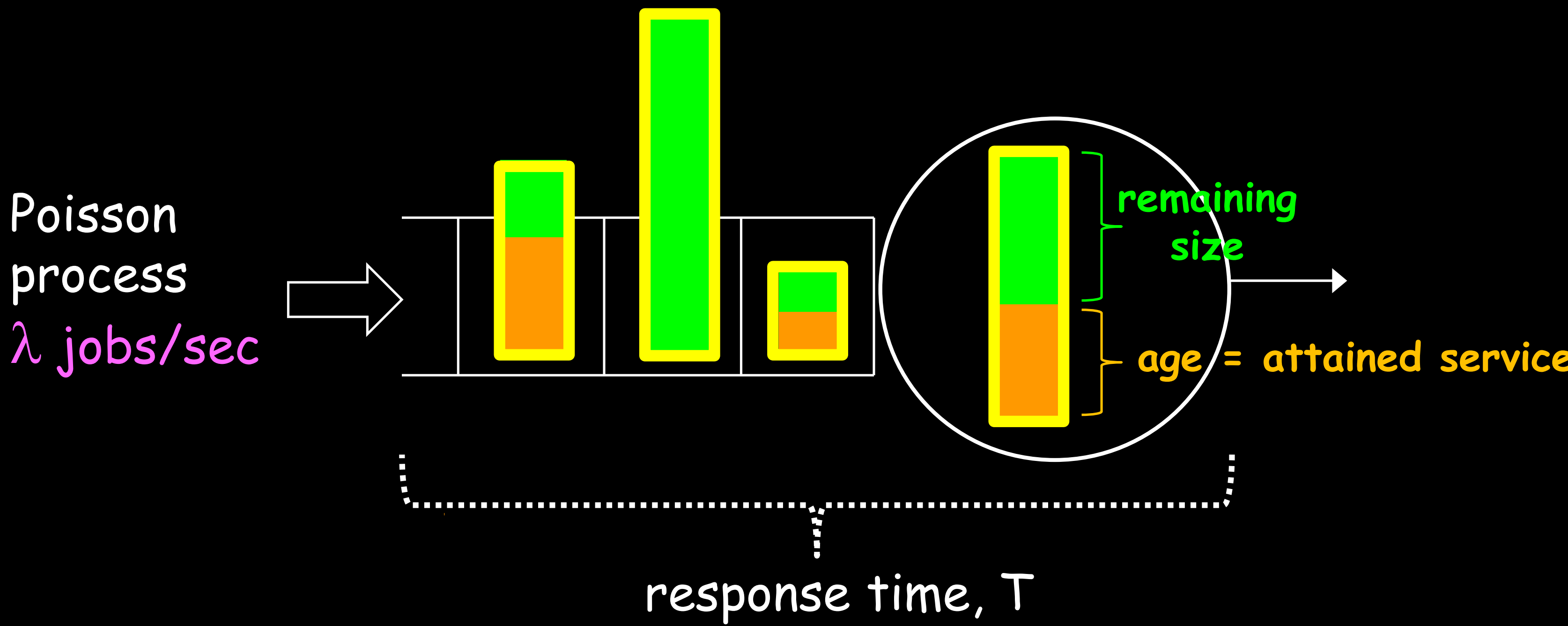


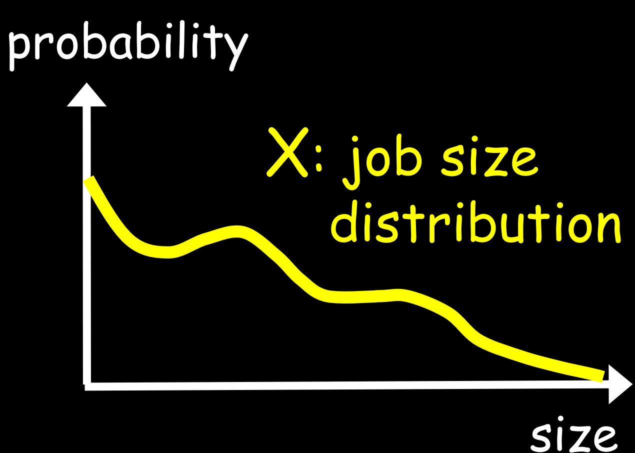
Basic Terminology



Scheduling Policy
(preempt-resume)

M/G/1 with Scheduling

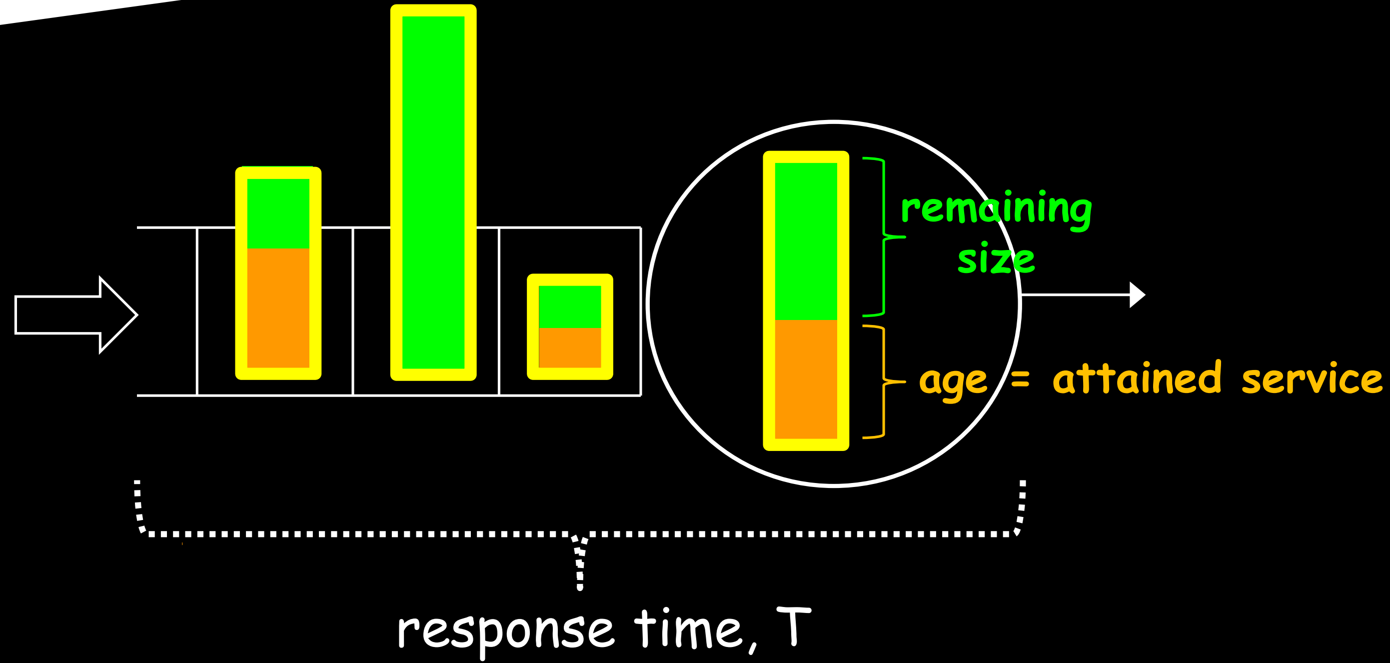




"Load" = fraction time server busy

$$\rho = \lambda \cdot E[X] < 1$$

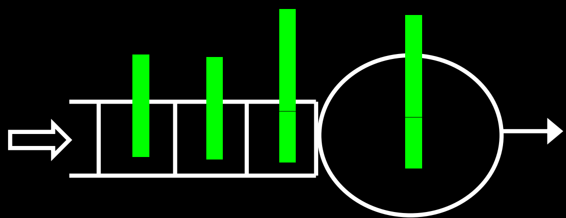
Q: What scheduling policy minimizes E[T]?



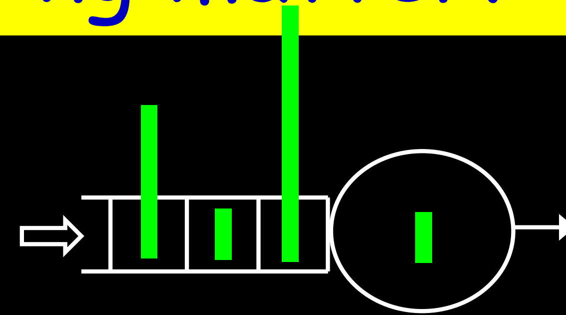
A: SRPT -Shortest Remaining Processing Time

[first M/G/1 analysis -- Schrage 1966]

How much does scheduling matter?



$C_X^2 = 1$

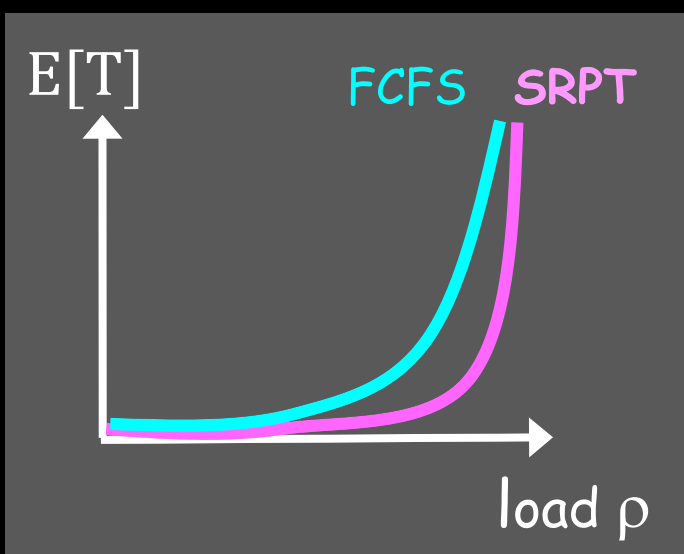


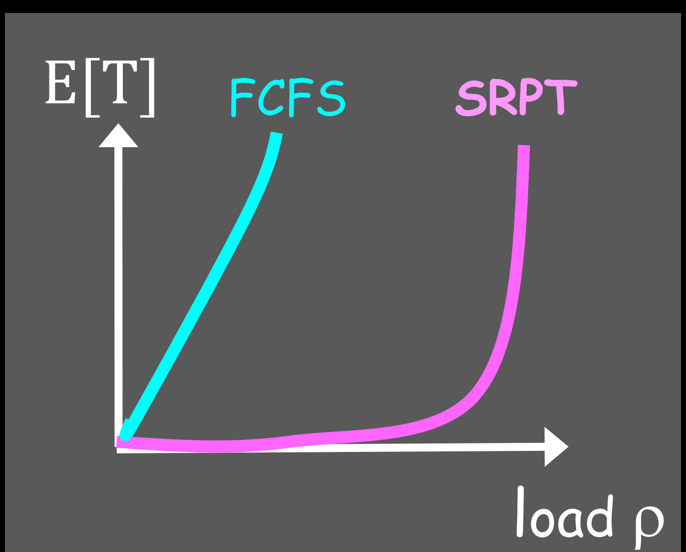
$C_X^2 = 100$

$$C_X^2 = \frac{Var(X)}{E[X]^2}$$

Low variability

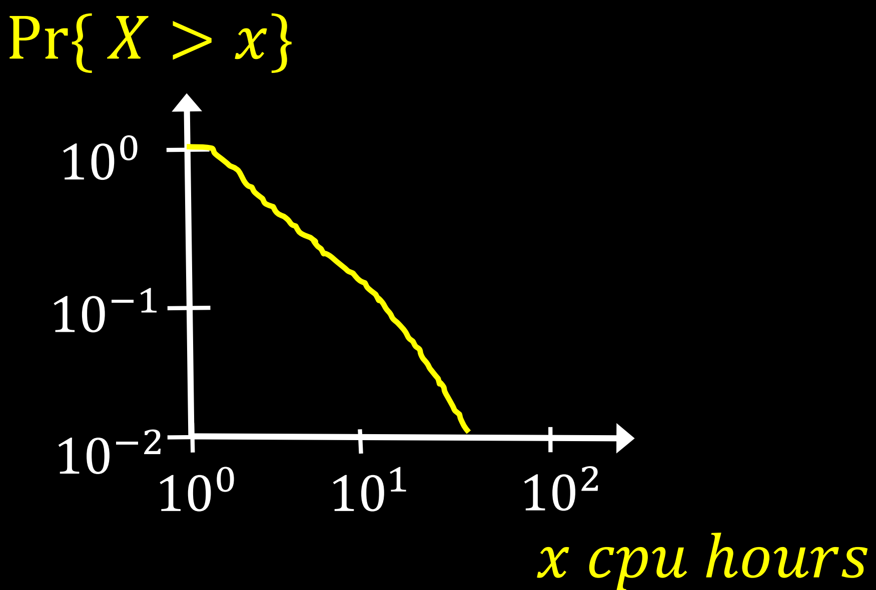
High variability





Empirical Job Size Distribution

UNIX jobs.
[Harchol-Balter, Downey – SIGMETRICS 1996]



$\underline{X = Job\ Size}$

$X \sim BoundedPareto(\alpha = 1.0)$

$C_X^2 = 50$

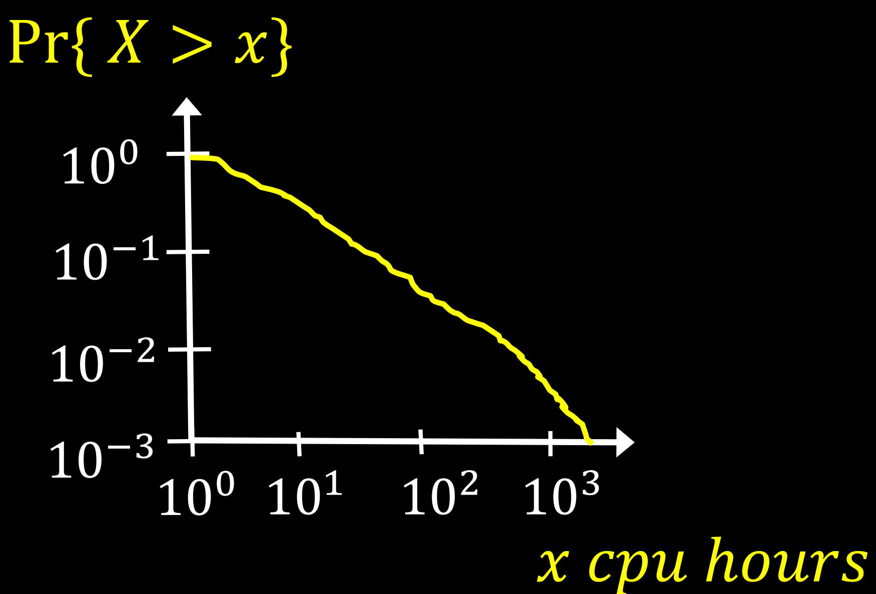
$Top\ 1\%\ of\ jobs = 50\%\ of\ load$

Upshot: Scheduling matters

Empirical Job Size Distribution

Borg Scheduler at Google

[Tirmazi, Barker, Deng, Haque, Qin, Hand, Harchol-Balter, Wilkes EUROSYS 2020]



$X = Job\ Size$

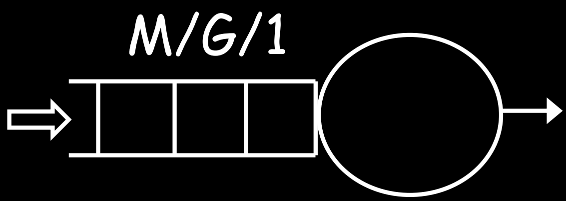
$X \sim BoundedPareto(\alpha = 0.69)$

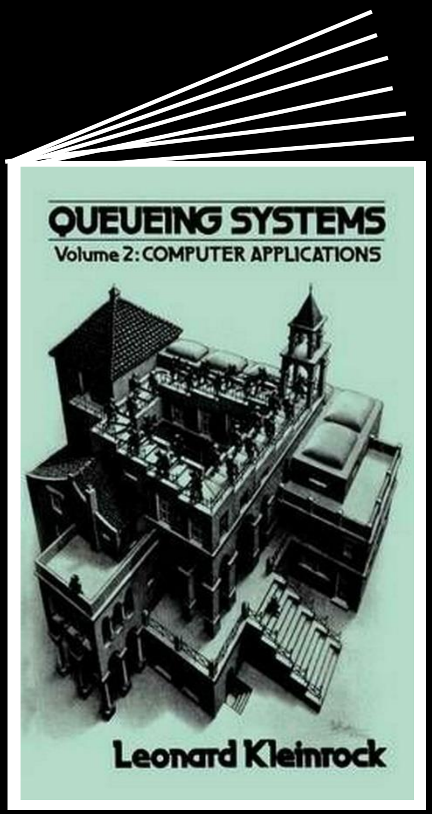
$C_X^2 = 23{,}000$

$Top\ 1\%\ of\ jobs = 99\%\ of\ load$

Upshot: Scheduling REALLY matters!

so FEW scheduling policies analyzable…





$$E[T(x)]^{FCFS} = \frac{\lambda\, E[X^2]}{2(1-\rho)} + x$$

$$E[T(x)]^{SRPT} = \frac{\lambda E[\min(X,x)^2]}{2(1-\rho_{\leq x})^2} + \int_{t=0}^{x} \frac{dt}{1-\rho_{\leq t}}$$

Similar response time formulas for: FB, PS, MLPS, PSJF, SJF, LCFS, PLCFS, NP-Prio, P-Prio.

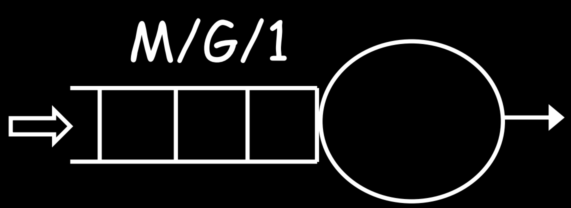
And that's basically it!

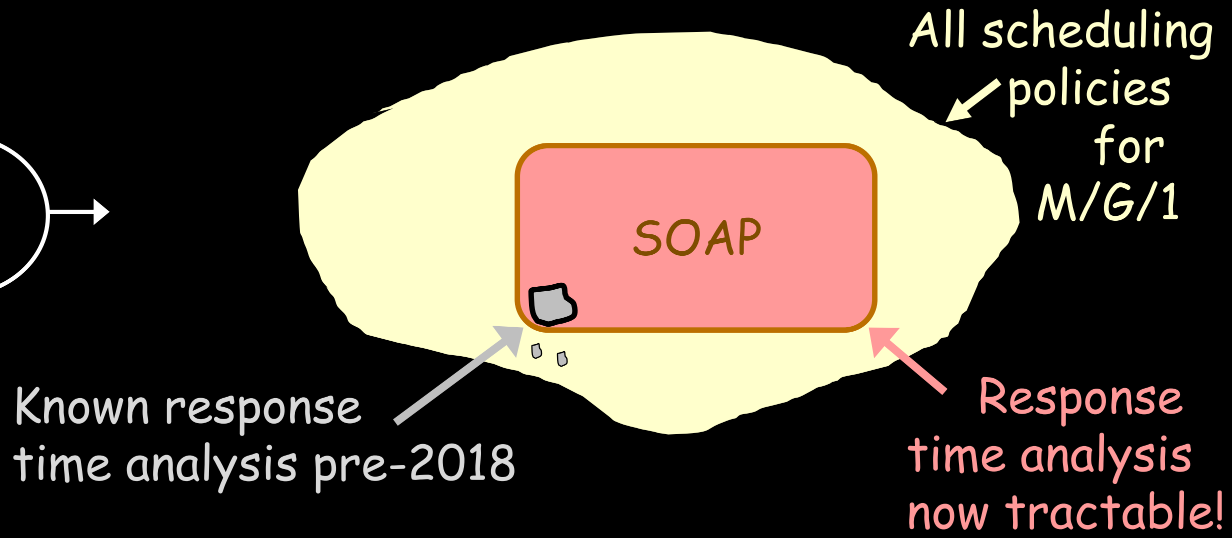
so MANY policies we can't analyze

Outline

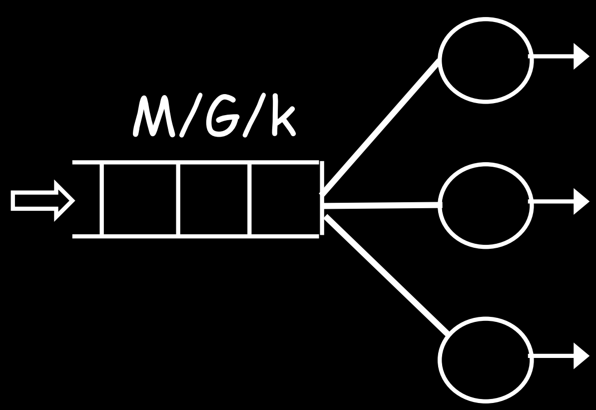
Stochastic scheduling breakthroughs in past 3 years

I.





II.



Scheduling in multi-server systems wide open:

- First bounds
- Optimality results

Papers relevant to this talk

Scully, Harchol-Balter, Scheller-Wolf – SIGMETRICS 2018

Grosof, Scully, Harchol-Balter – IFIP PERFORMANCE 2018

Scully, Harchol-Balter – ALLERTON 2018

Grosof, Scully, Harchol-Balter – IFIP PERFORMANCE 2019

Scully, Harchol-Balter, Scheller-Wolf – SIGMETRICS 2020

Scully, Grosof, Harchol-Balter – IFIP PERFORMANCE 2020

Scully, Grosof, Harchol-Balter- SIGMETRICS 2021

Grosof, Yang, Scully, Harchol-Balter- SIGMETRICS 2021

INFORMS '18 APS Finalist; Performance '18 Award;
Sigmetrics '19 Award; Sigmetrics '20 Award

Ziv Scully

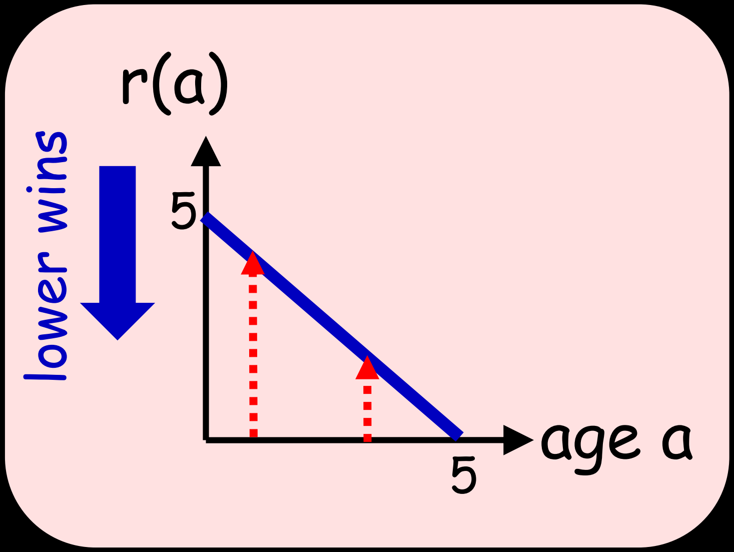
Isaac Grosof

<u>SOAP Policies</u>: all policies expressible via a rank function.

- ➢ Rank is a function of age
  (and the job's size or class)
- ➢ Always serve job of lowest rank
- ➢ FCFS tie-breaking

Example of classic SOAP policy:

SRPT

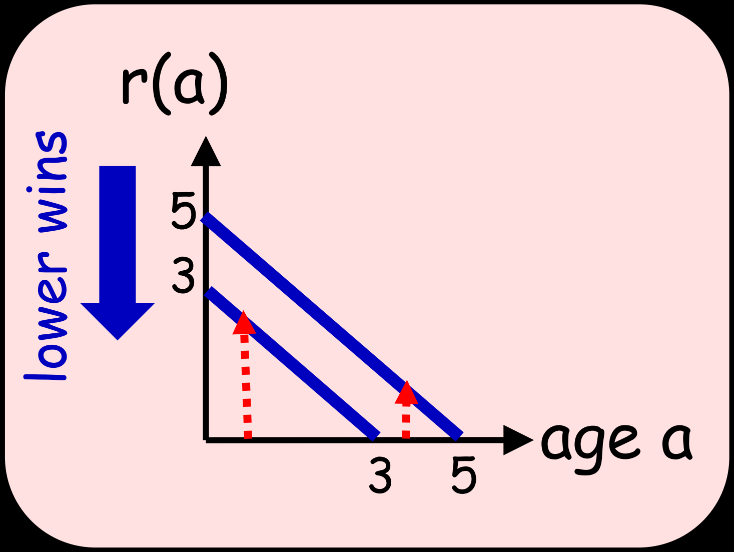


<u>SOAP Policies</u>: all policies expressible via a rank function.

- Rank is a function of age
  (and the job's size or class)
- Always serve job of lowest rank
- FCFS tie-breaking

Example of classic SOAP policy:

SRPT

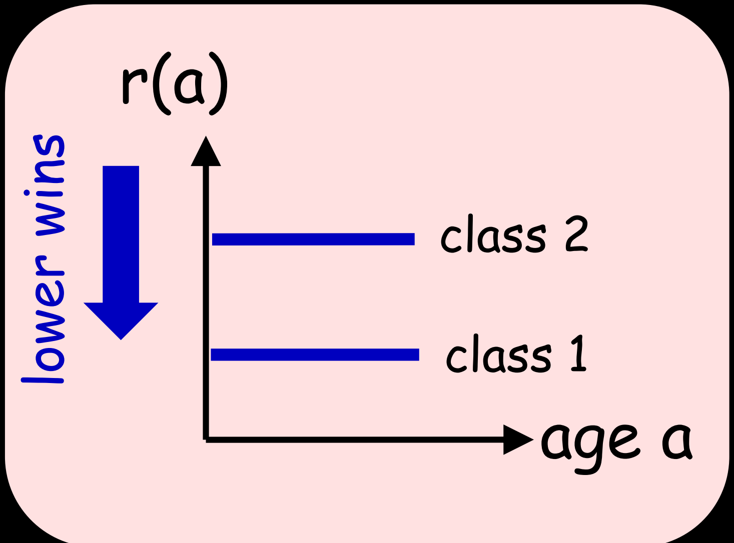


<u>SOAP Policies</u>: all policies expressible via a rank function.

- Rank is a function of age
  (and the job's size or class)
- Always serve job of lowest rank
- FCFS tie-breaking

Priority

Example of classic SOAP policy:

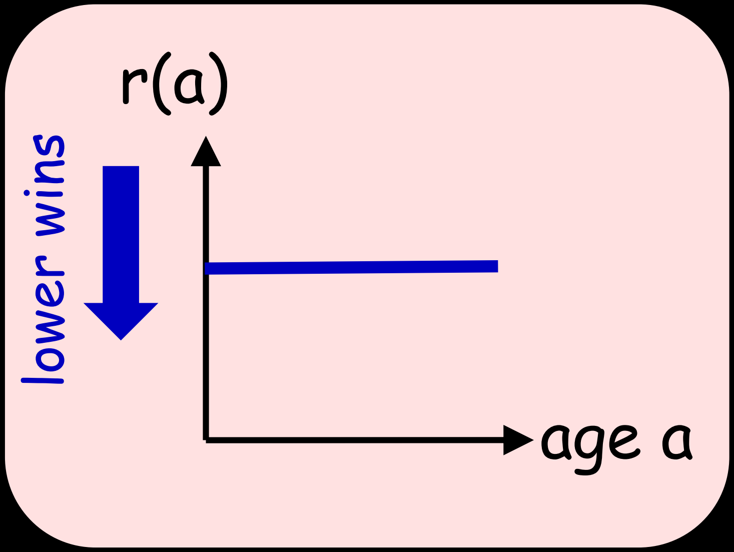


SOAP Policies: all policies expressible via a rank function.

- Rank is a function of age
  (and the job's size or class)
- Always serve job of lowest rank
- FCFS tie-breaking

FCFS

Example of classic SOAP policy:

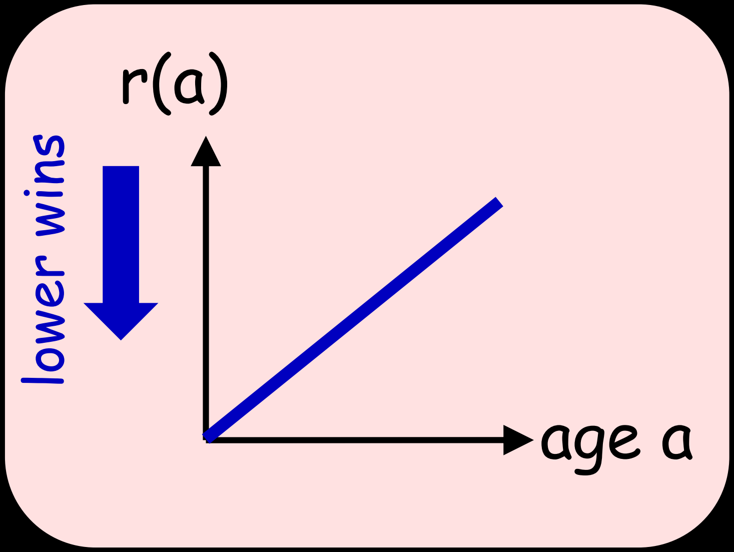


<u>SOAP Policies</u>: all policies expressible via a rank function.

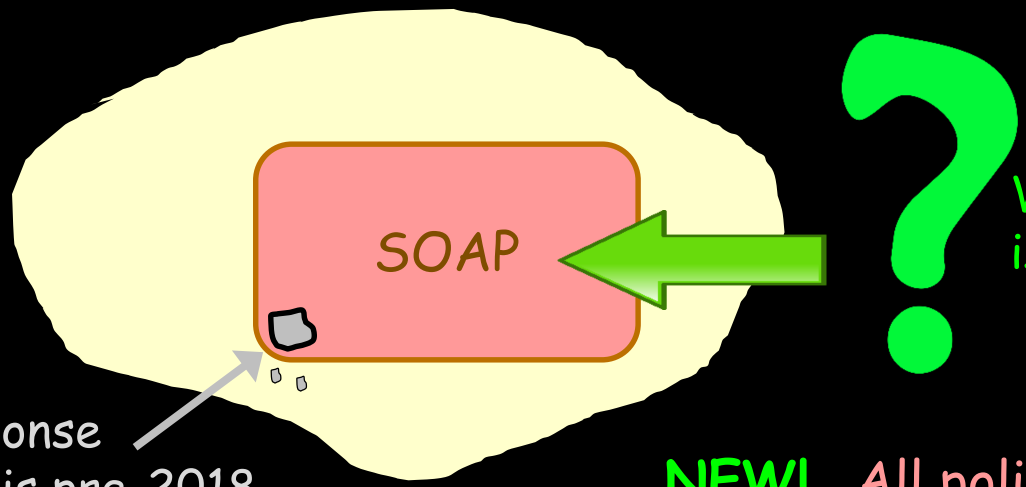
- Rank is a function of age
  (and the job's size or class)
- Always serve job of lowest rank
- FCFS tie-breaking

Example of classic SOAP policy:

LAS



All scheduling policies for M/G/1



What else
is in SOAP?

Known response
time analysis pre-2018

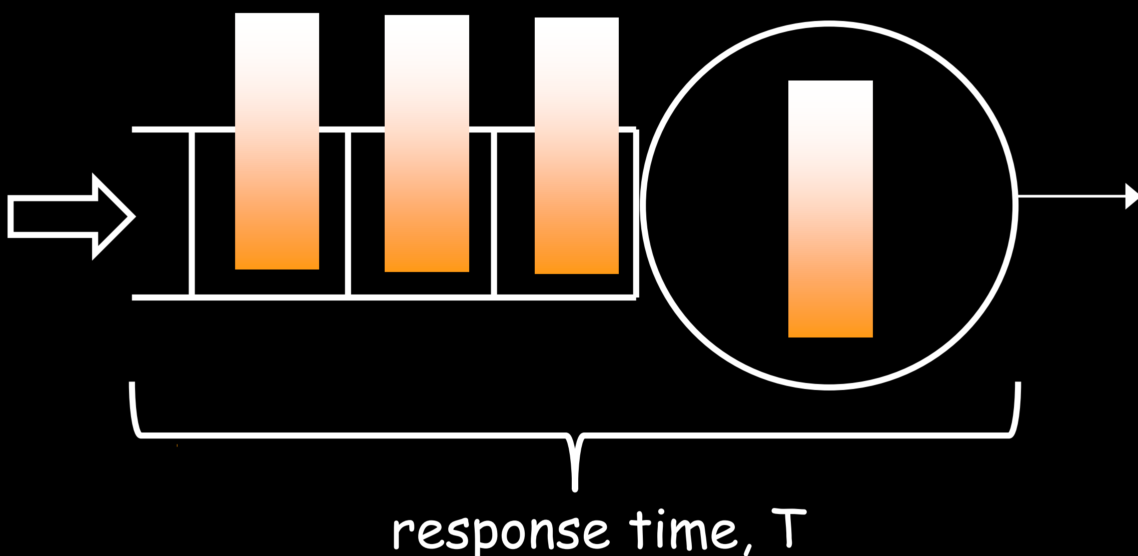
e.g., SRPT, FCFS, Prio, LAS

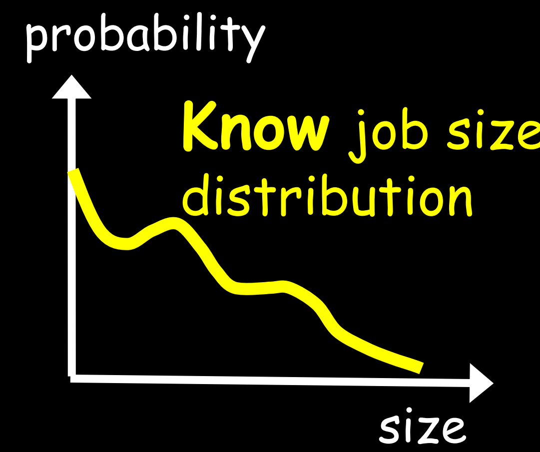
monotonic
rank functions

NEW! All policies with
non-monotonic
(or monotonic)
rank functions

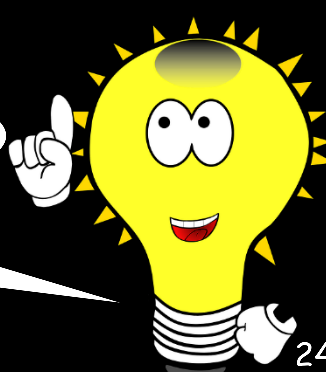
But why do
we care
about
non-monotonic?

Q: How should we schedule when don't know job size?





SERPT -- Shortest Expected Remaining Processing Time

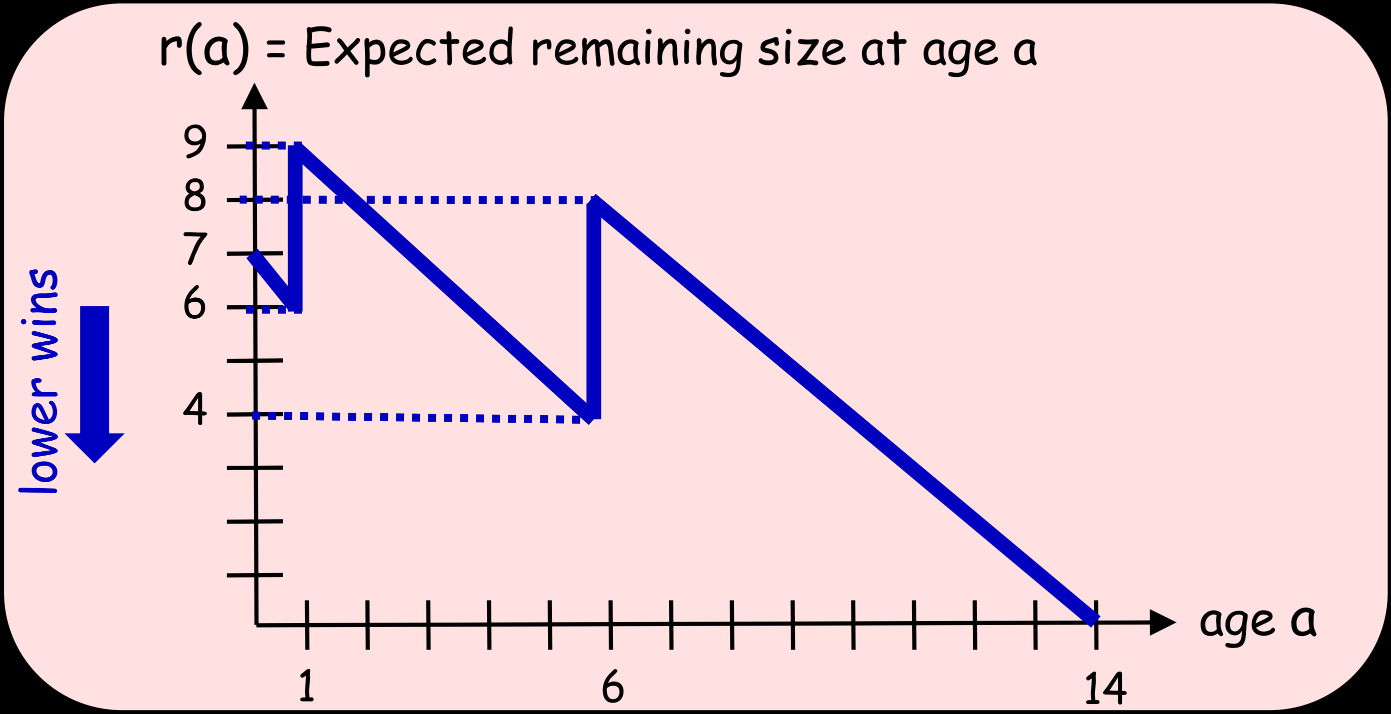


SERPT

Always run job with lowest rank

$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$

$$r(a) = E[X - a \mid X > a]$$



rank NOT monotonic

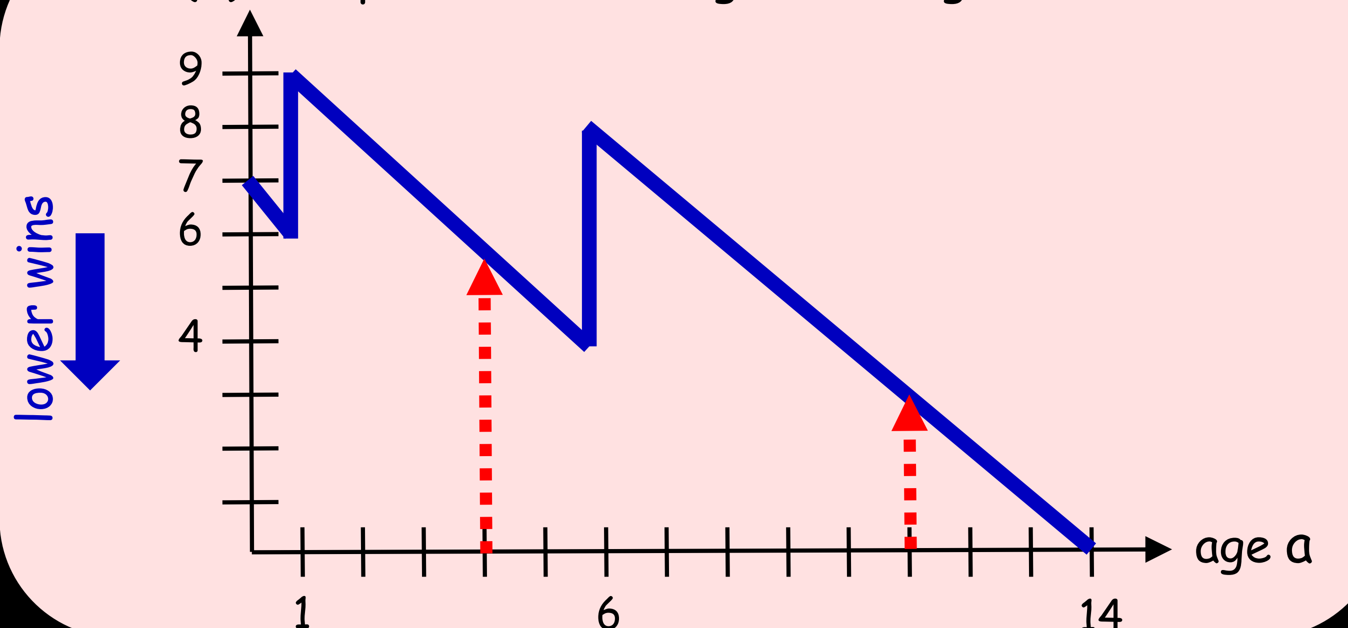
SERPT

Always run job with lowest rank

$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$

$$r(a) = E[X - a \mid X > a]$$

r(a) = Expected remaining size at age a



rank NOT monotonic

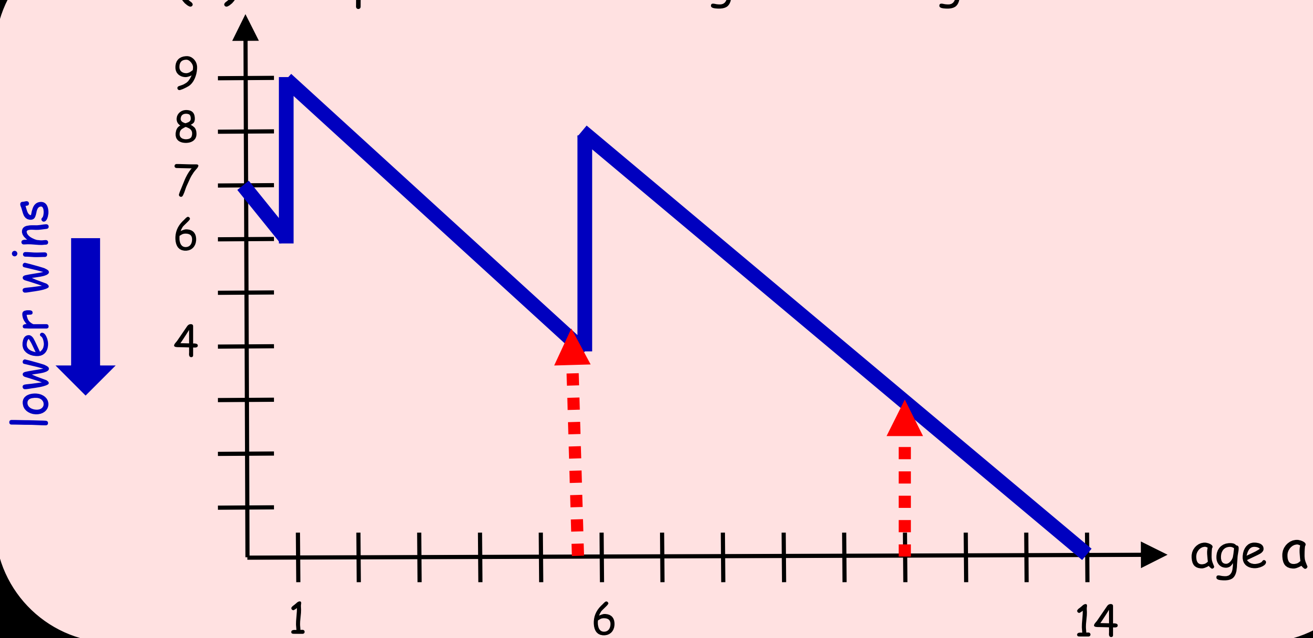
SERPT

Always run job with lowest rank

$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$

$$r(a) = E[X - a \mid X > a]$$

r(a) = Expected remaining size at age a



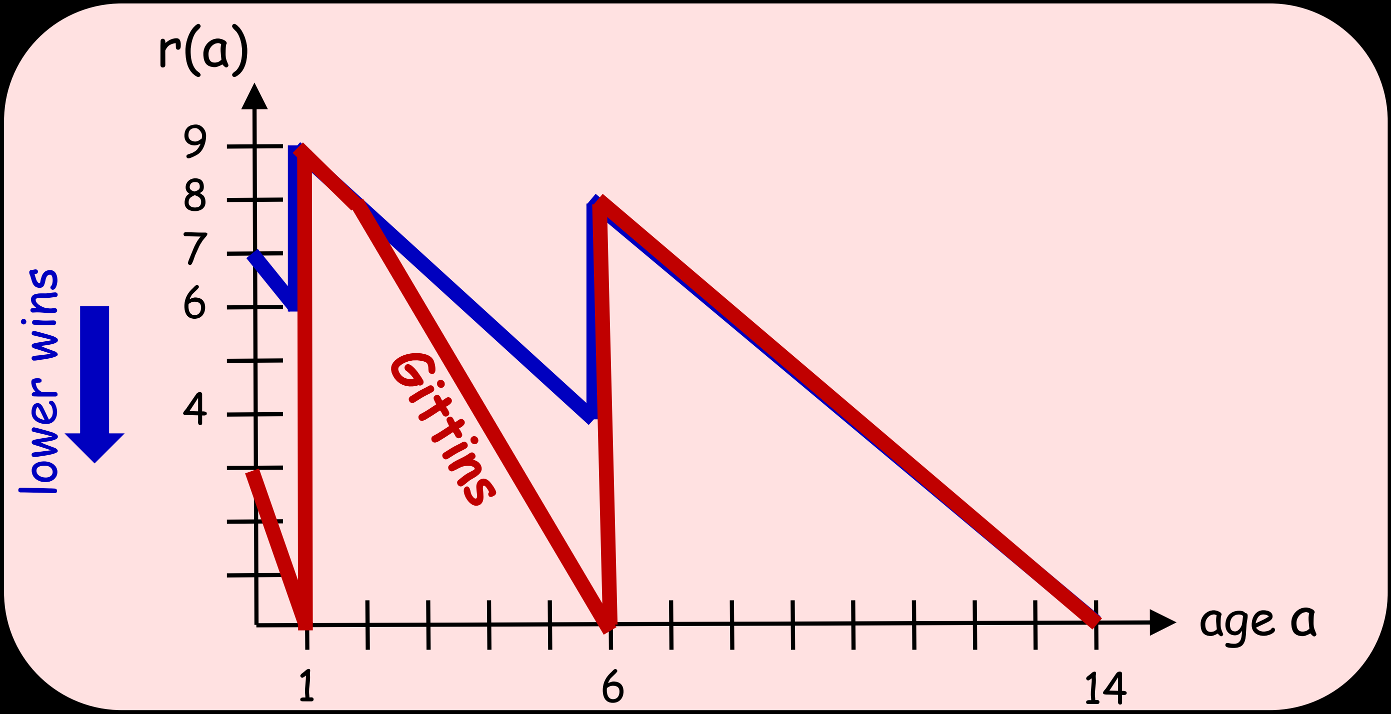
rank NOT monotonic

Gittins

Always run job with lowest rank

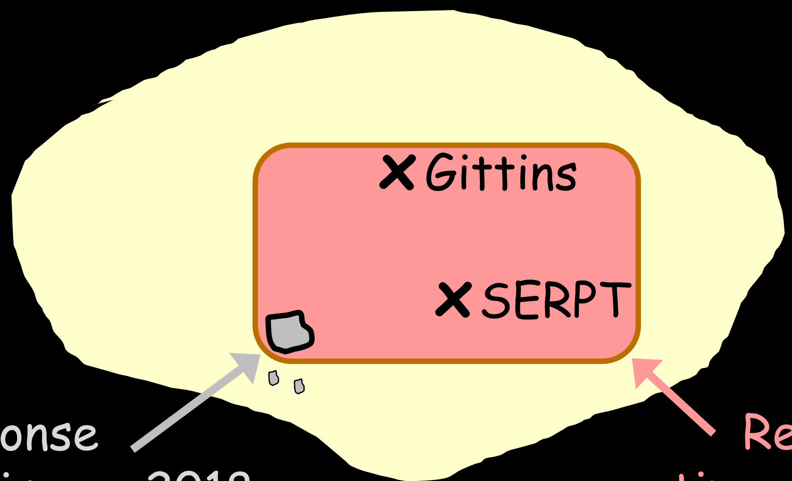
$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$

$$r(a) = \inf_{\Delta} \frac{E[\min\{X-a\,,\ \Delta\}\,|\,X>a]}{\Pr\{\,X\ \le a+\ \Delta\,|\ \ X>a\}}$$



rank NOT monotonic

All scheduling policies for M/G/1



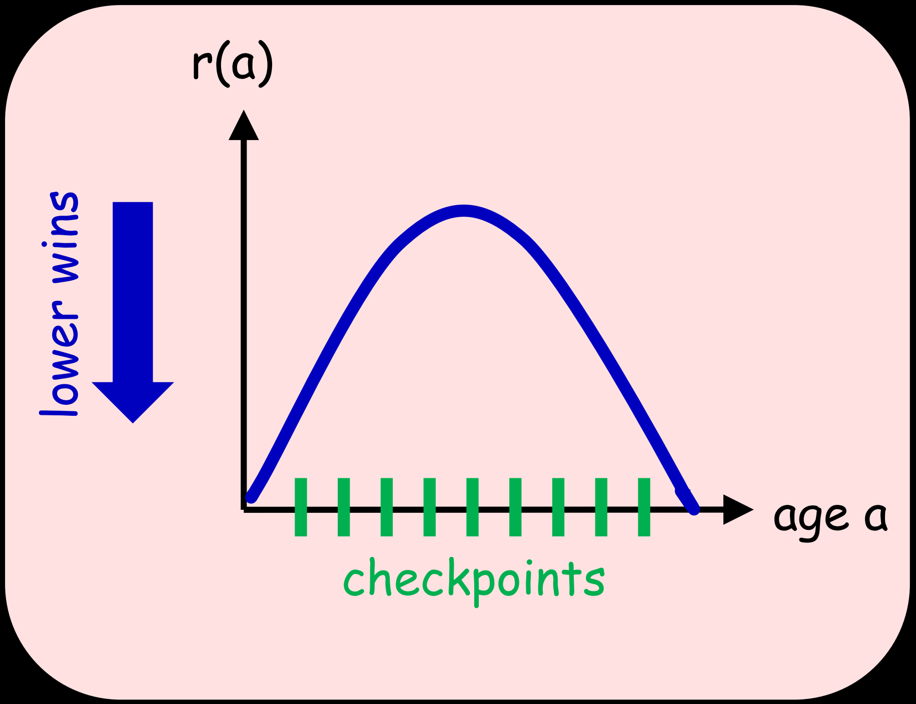
Known response
time analysis pre-2018

Response
time analysis
now tractable!

First response time anlaysis
of Gittins and SERPT in M/G/1
[Sigmetrics 2018 "SOAP" paper]

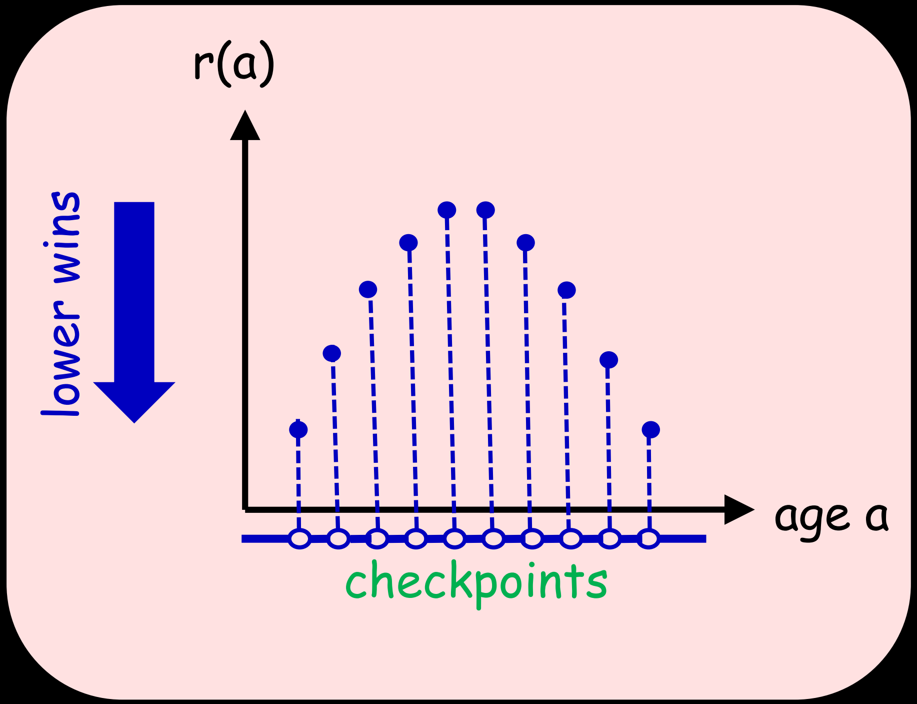
More SOAP policies

Any policy where preemption is limited to checkpoints



More SOAP policies

Any policy where preemption is limited to checkpoints



rank NOT monotonic

More SOAP policies

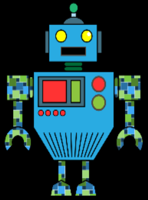
Mixed Classes

Humans (prio 1)

- Non-Preempt
- Unknown Size
- FCFS

vs.

Robots (prio 2)



- Preempt
- Known Size
- SRPT

Twist:
If $remsize(robot) < x_H$ *then robot has priority over un-started human.*

$$r_{Human}(a) = (-a\,,\ x_H\,)$$

$$r_{Robot(x)}(a) = (0\,,\ x - a)$$

All scheduling policies for M/G/1

✗ SERPT
✗ Gittins
✗ LimitedChkpt
✗ MixedPrio

Known response
time analysis pre-2018

monotonic rank functions

Response time
analysis now tractable!

All policies with **non-monotonic**
or monotonic
multi-dimensional rank functions

Given:
any rank function

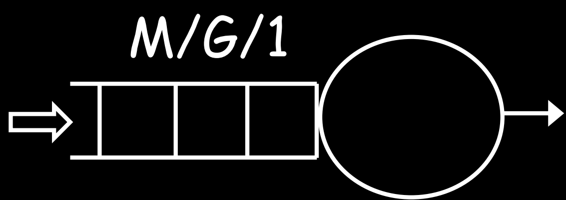
[Sigmetrics '18]

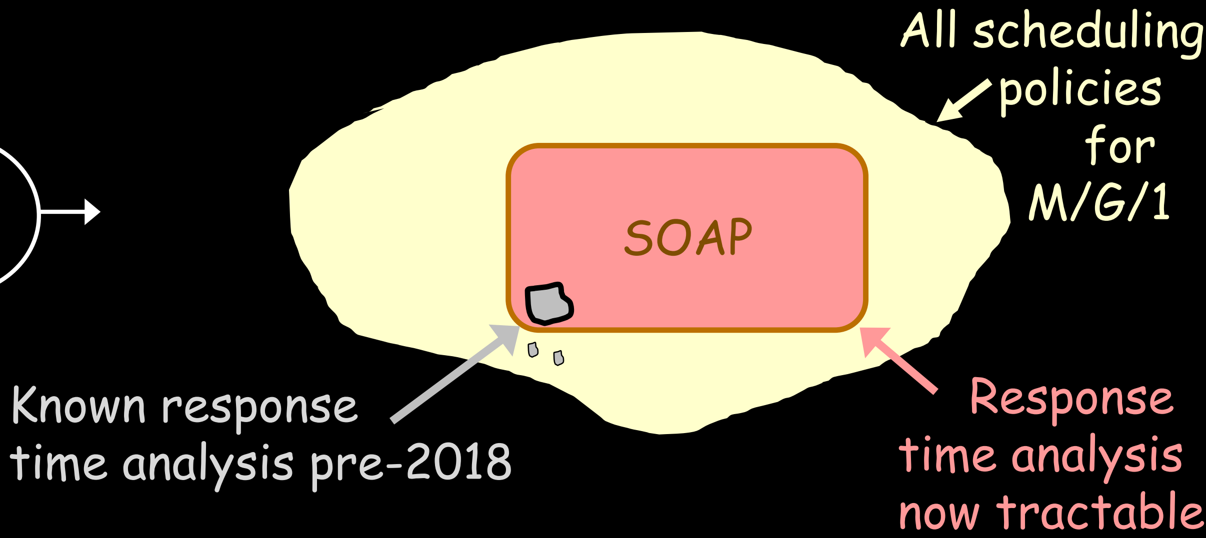
Closed-form
response time
(mean & transform)

Outline

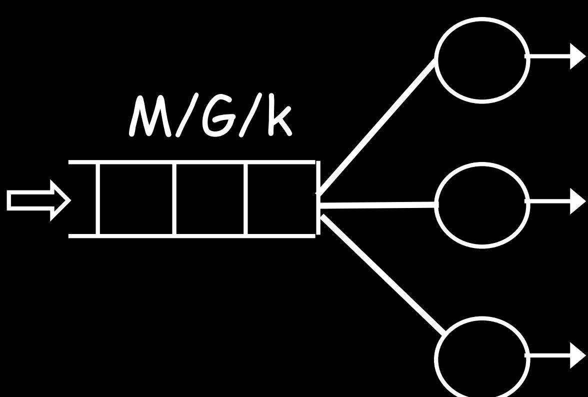
Stochastic scheduling breakthroughs in past 3 years

I.





II.

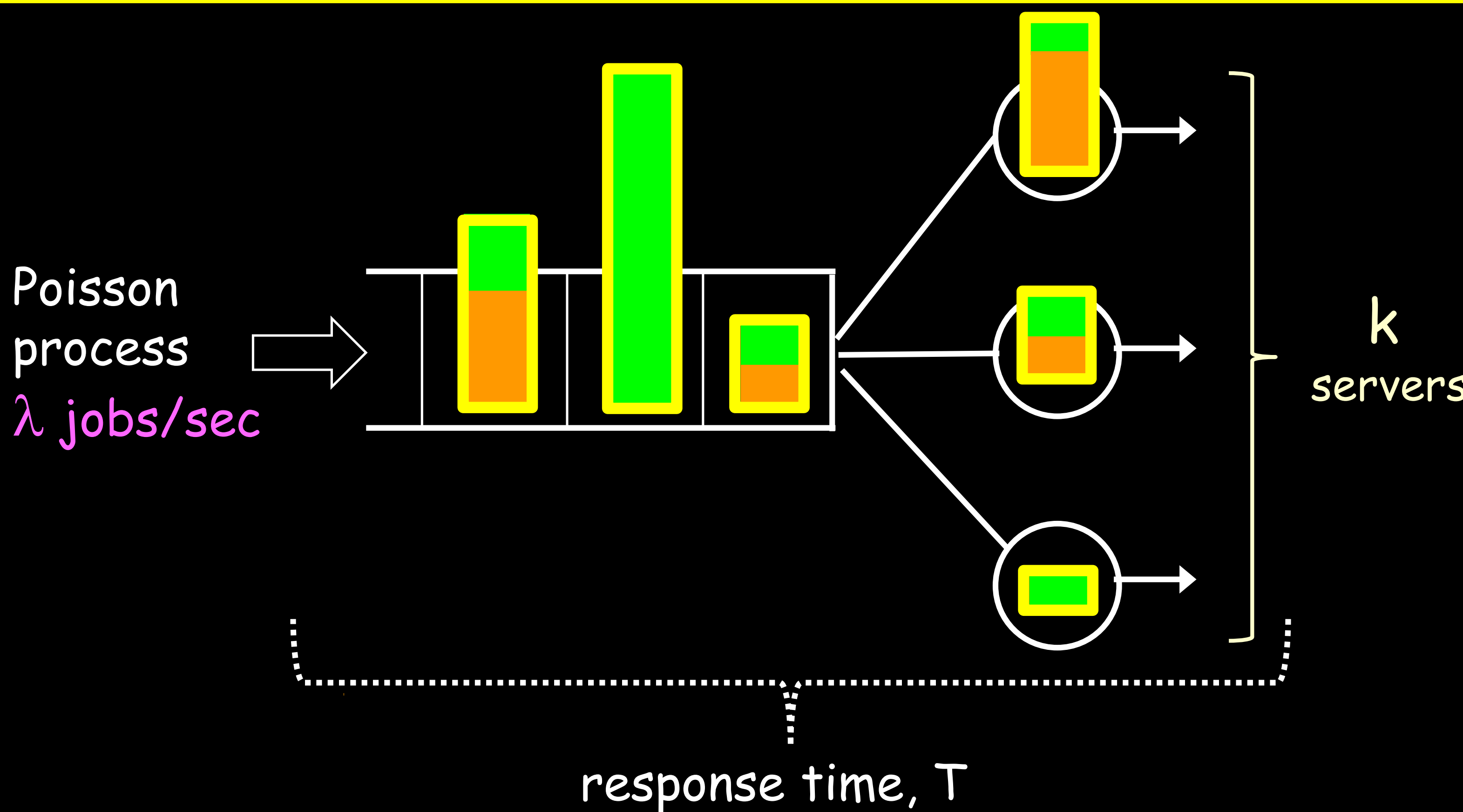


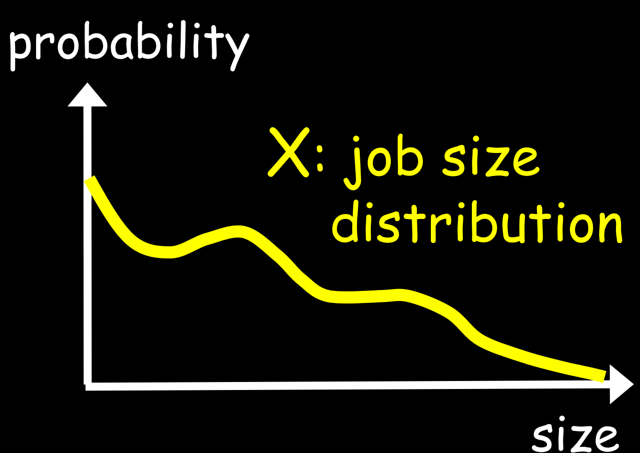
Scheduling in multi-server systems wide open:

- ❑ First bounds
- ❑ Optimality results

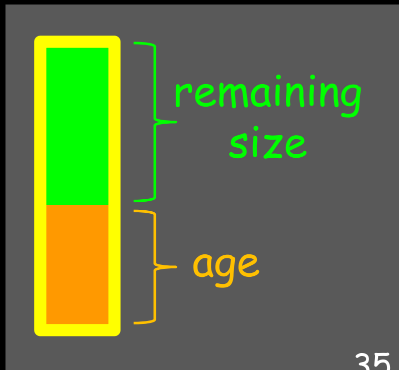
(start by assuming known sizes)

Multi-server system: M/G/k

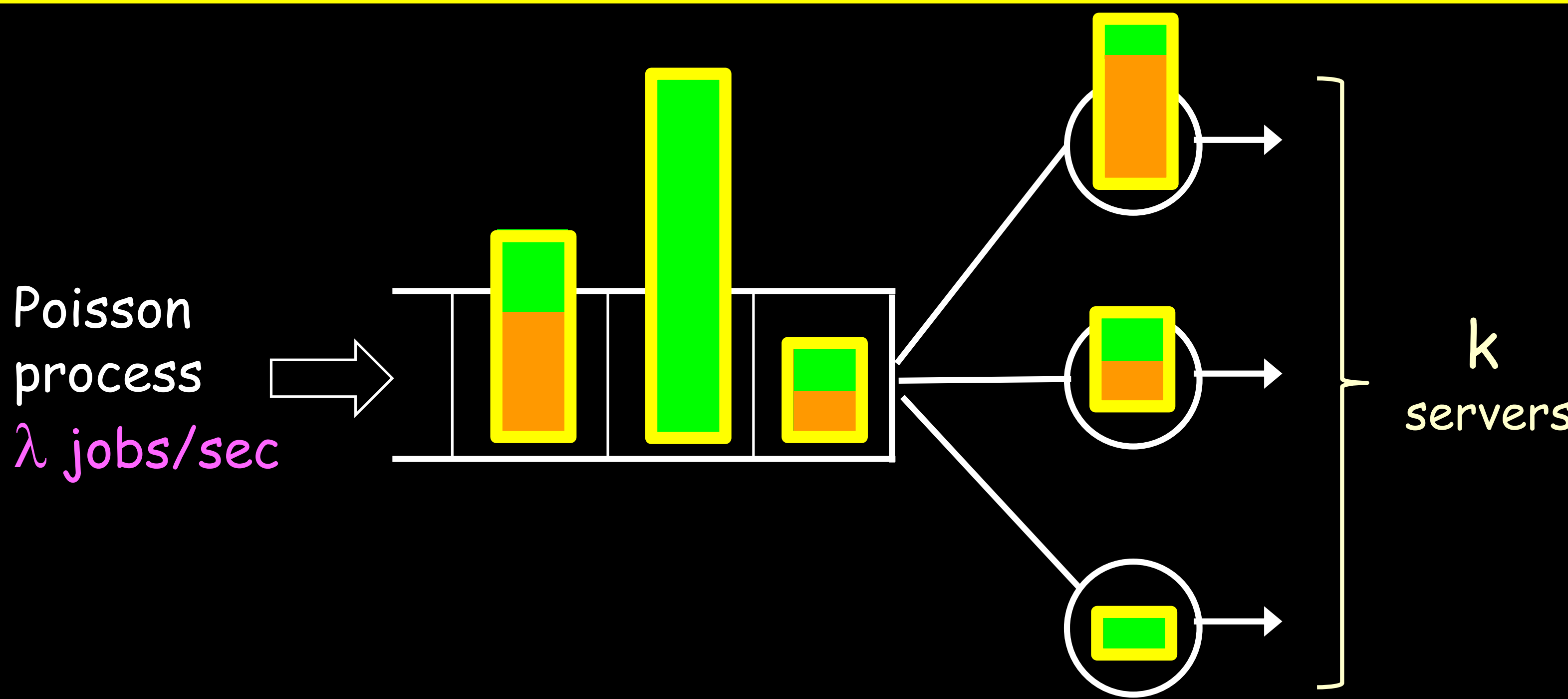




Assume job's size is known when it arrives!



Multi-server system: M/G/k



Q: How should we schedule to minimize $E[T]$?
(job sizes known)

SRPT-k ?

SRPT-k is FAR from OPT in worst-case

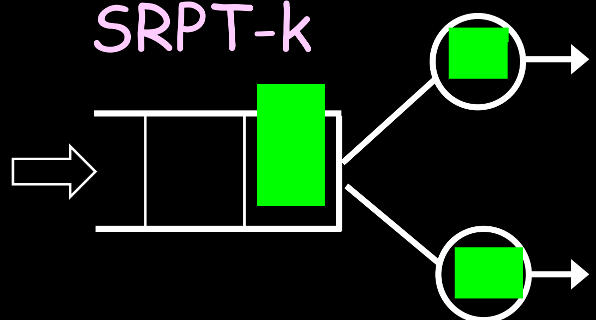
<u>Theorem</u>: [Leonardi, Raz 1997]

\# arrivals!

$$Competitive\ Ratio \leq log\left(\min\left(\boxed{\frac{n}{k}}, \boxed{\frac{Max\ job\ size}{Min\ job\ size}}\right)\right)$$

ratio can be high!

SRPT-k

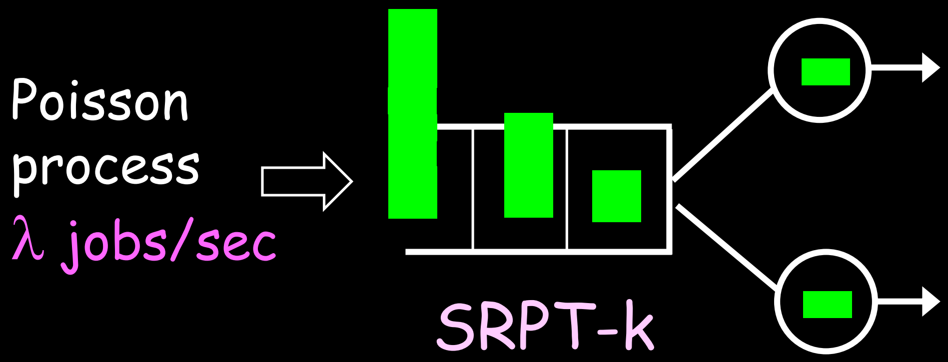


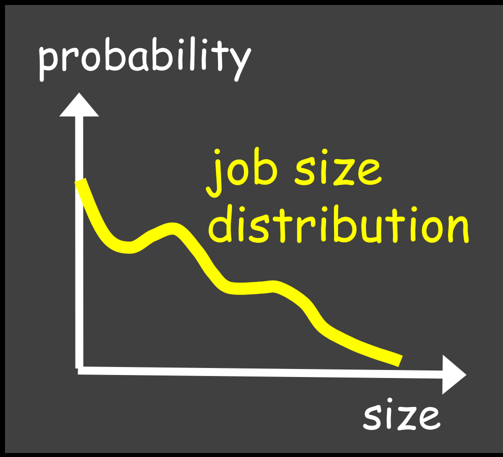
Poor bin-packing!

... and no other policy does better

CLOSED

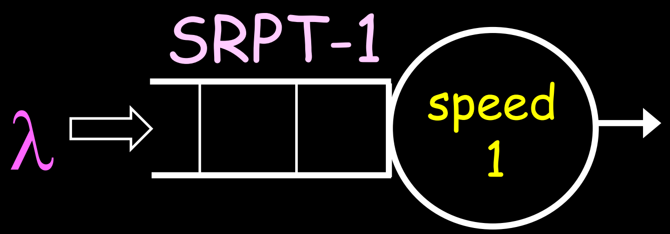
but maybe SRPT-k is not bad in M/G/k (stochastic) setting?



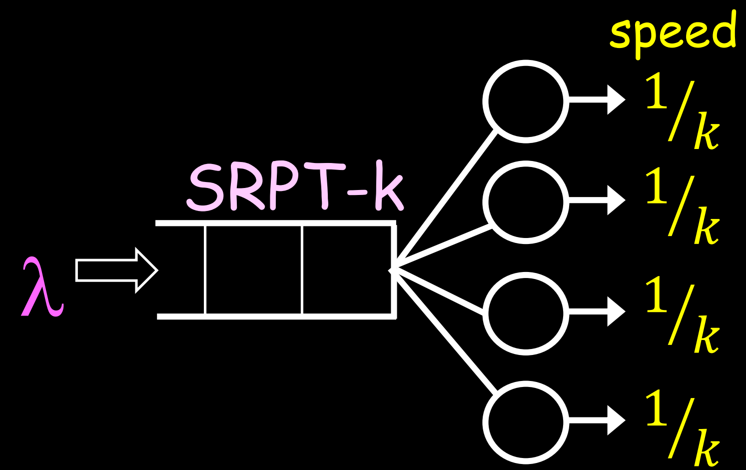


State-of-the-art for M/G/k scheduling mostly non-existent …

New approach! [Performance '18]



$\rho = \lambda \cdot E[X] =$ frac. of time server is busy



$\rho = \lambda \cdot E[X] =$ avg. frac. of servers busy

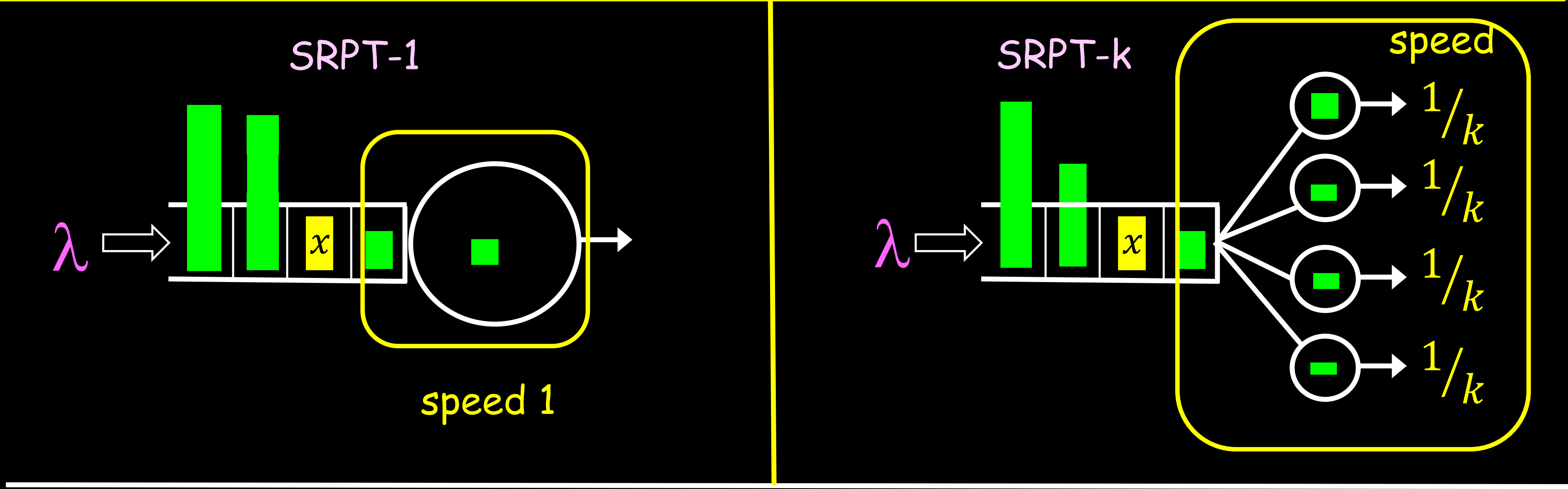
$$E[T]^{OPT-1} \leq E[T]^{OPT-k}$$

We show 2 results:

1) <u>First Bound</u>: $E[T]^{SRPT-k} \leq E[T]^{SRPT-1} + \frac{2}{\lambda} k \ln\left(\frac{1}{1-\rho}\right)$

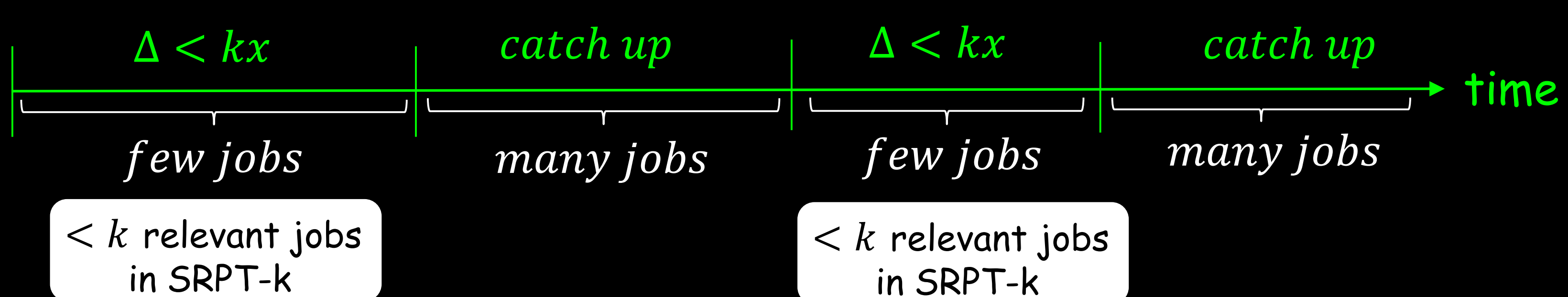
2) <u>Optimality</u>: $\lim_{\rho \to 1} \frac{E[T]^{SRPT-k}}{E[T]^{SRPT-1}} = 1$

Proof Sketch



- Show $RelevantWork(x)$ is similar in SRPT-1 and SRPT-k

$$\Delta = E[RelWork(x)]^{SRPT-k} - E[RelWork(x)]^{SRPT-1}$$



First response time bound for SRPT-k

$$E[RelWork(x)]^{SRPT-k} - E[RelWork(x)]^{SRPT-1} \le kx$$

First bound

$$E[T]^{SRPT-k} \le E[T]^{SRPT-1} + \boxed{\frac{2}{\lambda} k \ln\left(\frac{1}{1-\rho}\right)}$$

$||$ [Lin, Wierman, Zwart 2011]

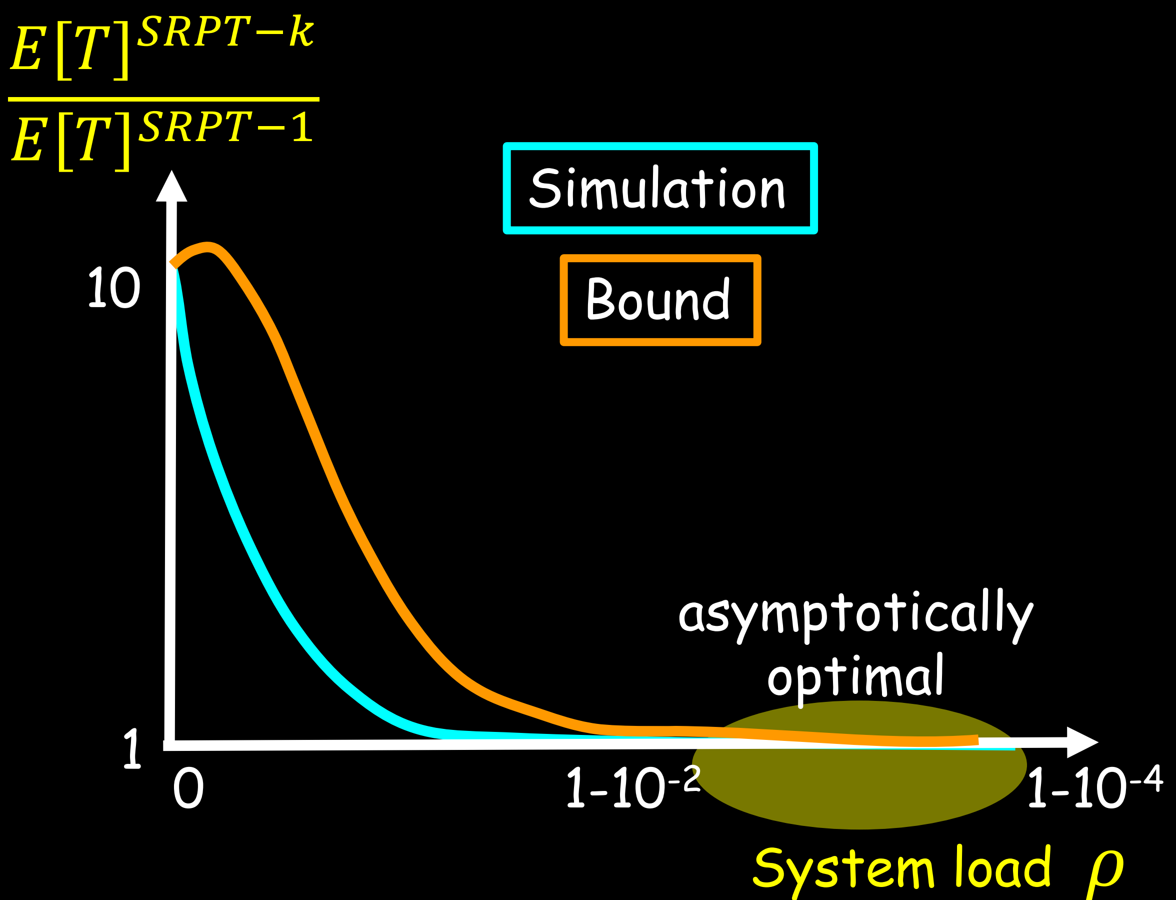
(assuming ≈ finite variance)

$$\boxed{o\left(E[T]^{SRPT-1}\right) \; as \; \rho \to 1}$$

Optimality result

$$\frac{E[T]^{SRPT-k}}{E[T]^{SRPT-1}} \to 1 \quad as \;\; \rho \to 1$$

Bound versus Simulation



[Performance '18]

Similar analysis for wide many M/G/k scheduling policies:

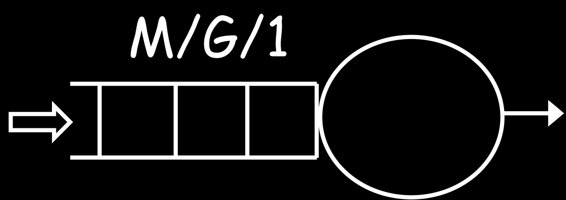
→ SRPT-k
→ PSJF-k
→ FB-k
→ RS-k

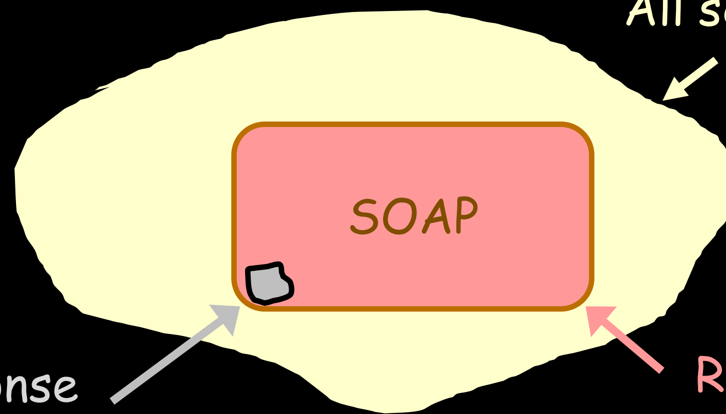
X~Uniform(0, 1) , k = 10 servers

Outline

Stochastic scheduling breakthroughs in past 3 years

I.



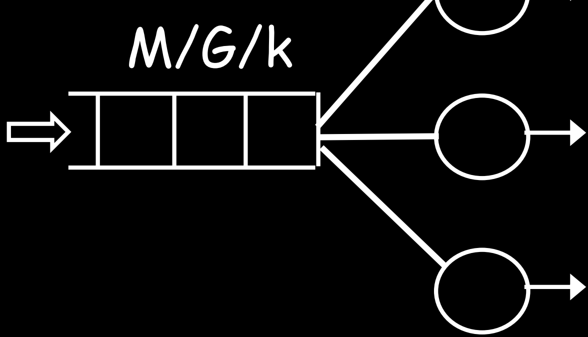


All scheduling policies for M/G/1

Known response time analysis pre-2018

Response time analysis now tractable!

II.

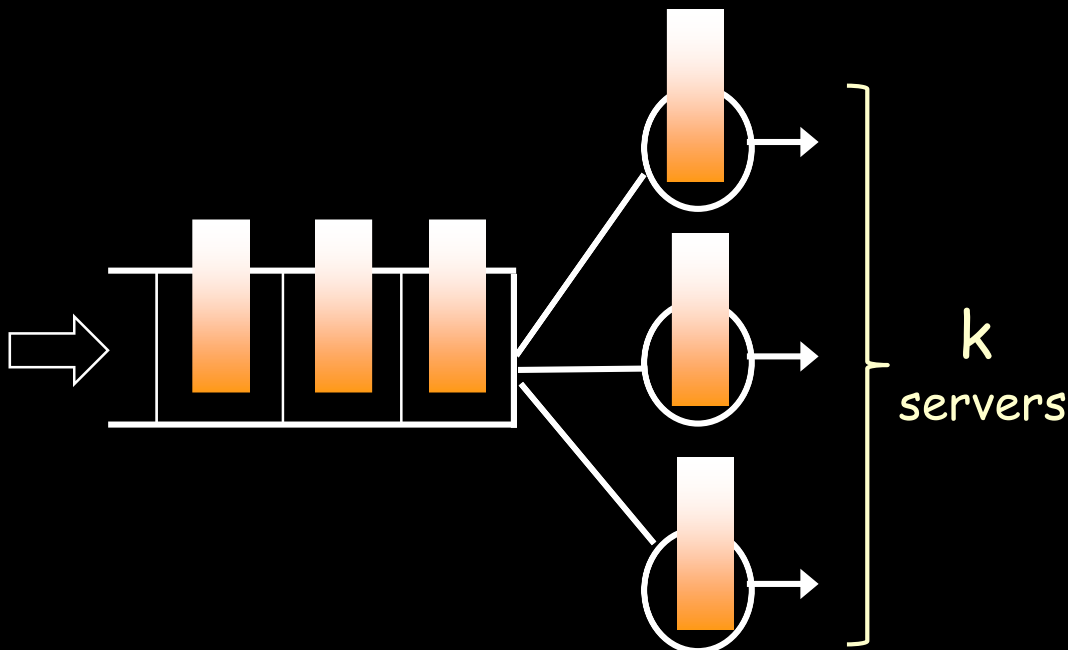


Scheduling in multi-server systems wide open:

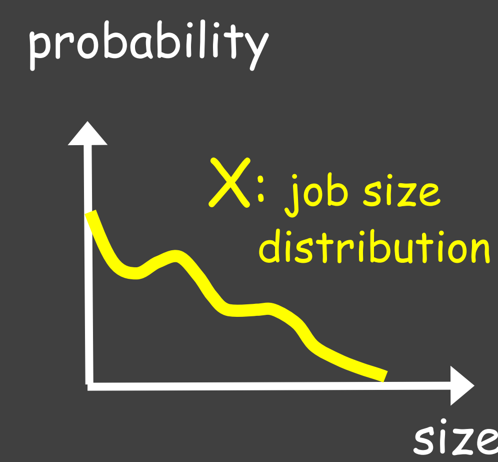
- First bounds
- Optimality results

Done with case where know size. What if don't know size?

Multi-server: Size Unknown [Sigmetrics 21]



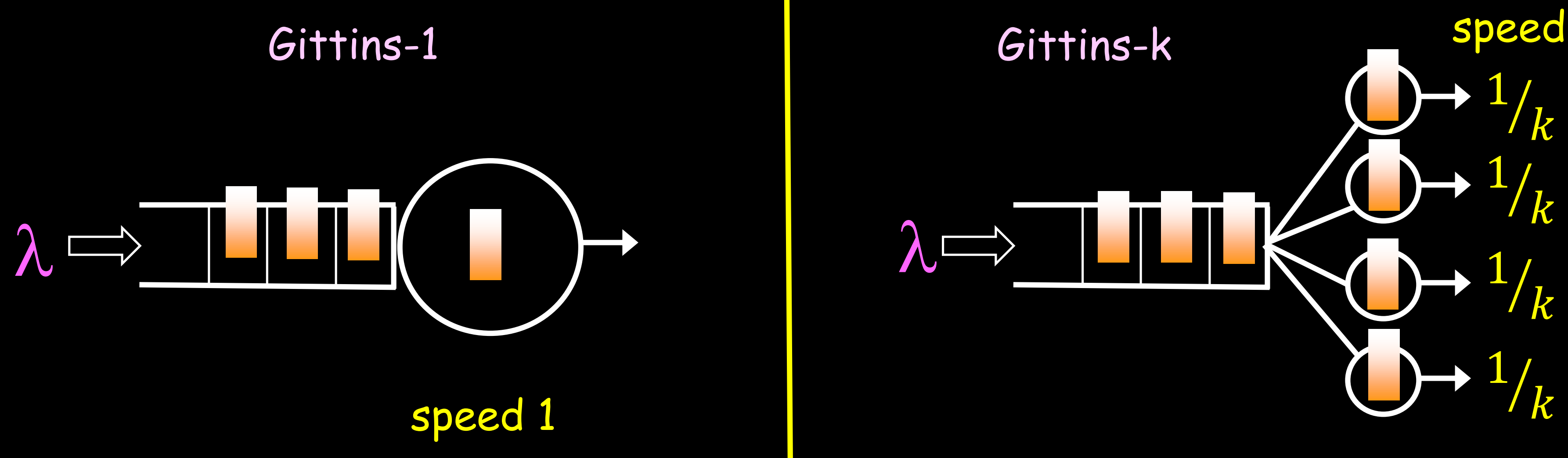
Job size distribution is known



Q: What scheduling policy makes sense here?

Gittins-k?

Gittins-k for M/G/k [Sigmetrics 21]



We show 2 results:

1) First Bound:

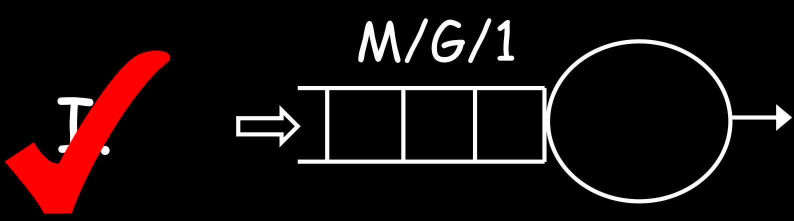
$$E[T]^{Gittins-k} \leq E[T]^{Gittins-1} + (k-1)E[X]\left(\ln\frac{1}{1-\rho} + \ln\frac{E[X^2]}{E[X]^2} + 4.9\right)$$

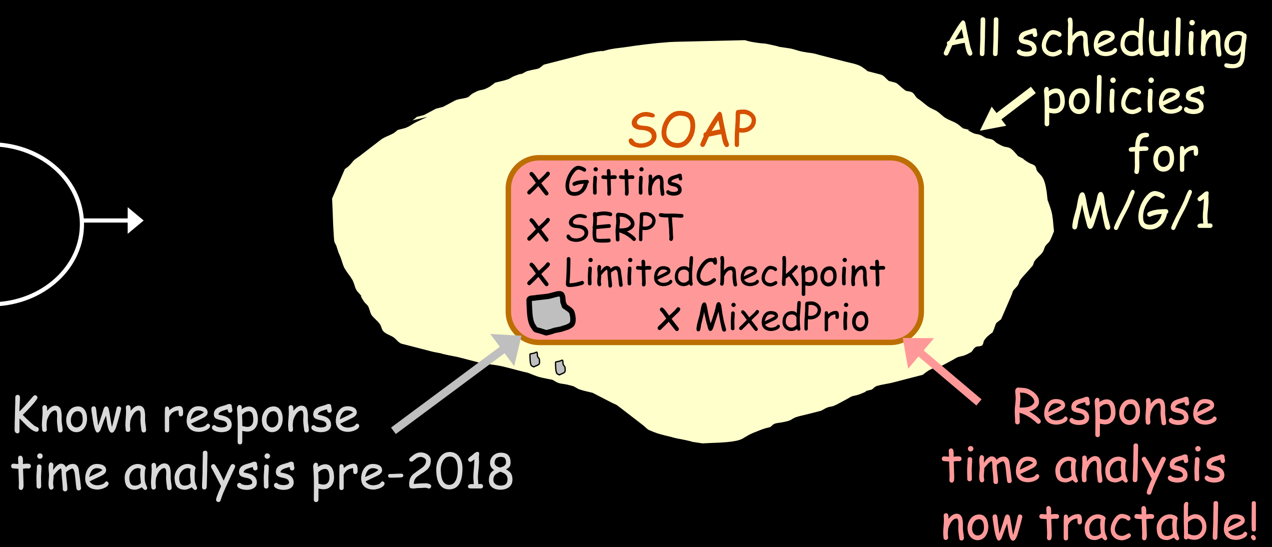
$$o\left(E[T]^{Gittins-1}\right)\ as\ \rho \to 1$$

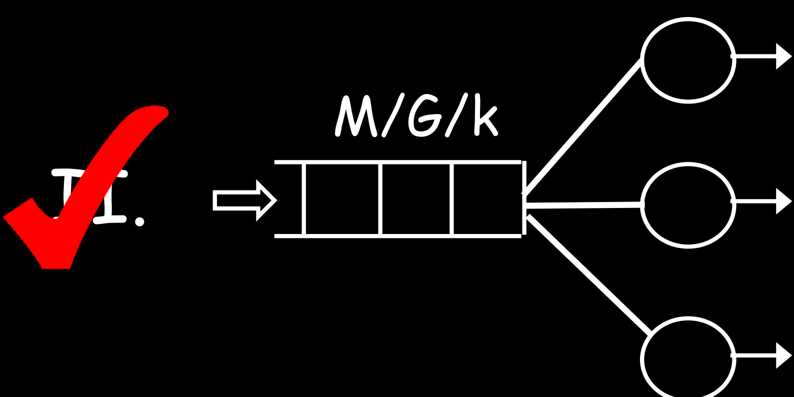
2) Optimality: $\lim_{\rho\to 1} \frac{E[T]^{Gittins-k}}{E[T]^{Gittins-1}} = 1$

Summary

Stochastic scheduling breakthroughs in past 3 years







<u>Scheduling in multi-server systems wide open</u>:

- ❑ First bounds
- ❑ Optimality results

SRPT-k,
PSJF-k,
RS-k,
FB-k,
Gittins-k

Open problems on stochastic scheduling…

Harchol-Balter. "Open problems in queueing theory inspired by datacenter computing." *Queueing Systems* , 97(1), 2021, pp. 3--37.